(12) UK Patent Application (19) GB (11) 2 369 537 (13) A

(43) Date of A Publication 29.05.2002

(21) Application No 0028616.1

(22) Date of Filing 24.11.2000

(71) Applicant(s)
Guang Yang
M.Z.S Main Building, University of Glasgow,
GLASGOW, G12 8QQ, United Kingdom

(72) Inventor(s)
Guang Yang

(74) Agent and/or Address for Service
Guang Yang
M.Z.S Main Building, University of Glasgow,
GLASGOW, G12 8QQ, United Kingdom

(51) INT CL[7]
H04L 29/06

(52) UK CL (Edition T)
H4P PPG

(56) Documents Cited
GB 2338870 A GB 2333670 A
WO 99/33211 A1

(58) Field of Search
UK CL (Edition S) H4P PPG
INT CL[7] H04L 29/06
ONLINE : EPODOC, JAPIO, WPI.

(54) Abstract Title

A web server with dynamic remote extensions

(57) A web server with dynamic remote extension is a computer network for DHCP (Dynamic Host Configuration Protocol) users to provide web contents over the Internet. Different from virtual host mechanism, these web contents reside on these extensions rather on the server. Every remote extension is an independent web host to publish its web contents. The web server works as gateway or agent to deal with incoming requests and to relay these request to its remote extensions. After fetching data from a specified extension, it sends the result back to requesters. The web server contains an extension name registration which stores the references and exported objects of live extensions in runtime. These extensions do not have static IP address configured for DNS and can only be registered themselves with the web server by exporting their remote objects that encapsulate host properties, web contents reading behaviours and communication events during their runtime. This invention relates to a dynamic web host construction without DNS resolving of IP address. It creates a new mechanism for the name resolving in the terms of binding the name with exported object of the web host in the runtime.

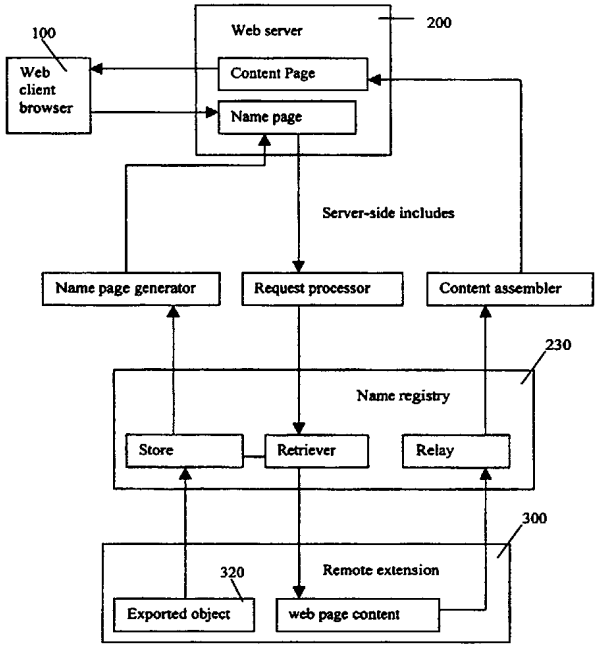

Fig. 4 illustrates the architecture and data flow of the web server network.

GB 2 369 537 A

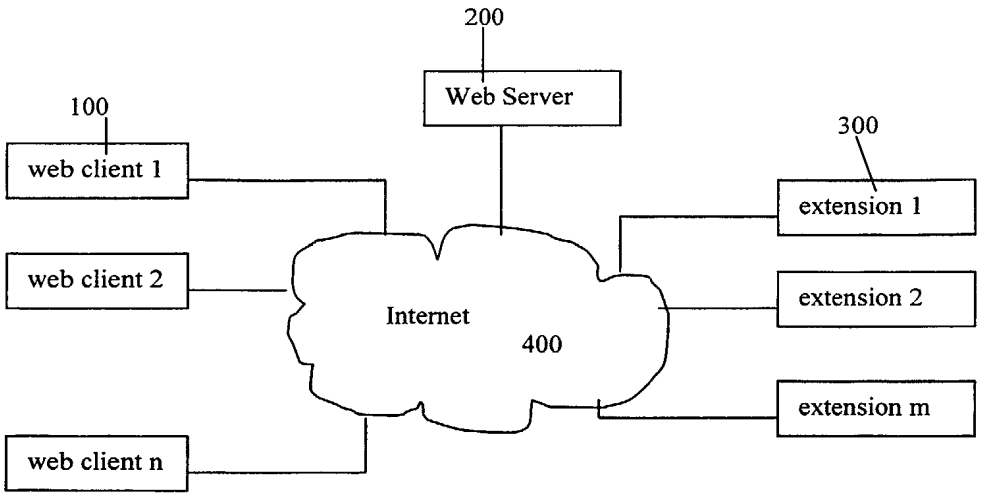

Fig. 1. illustrates physical connection of the network in accordance with the inventing

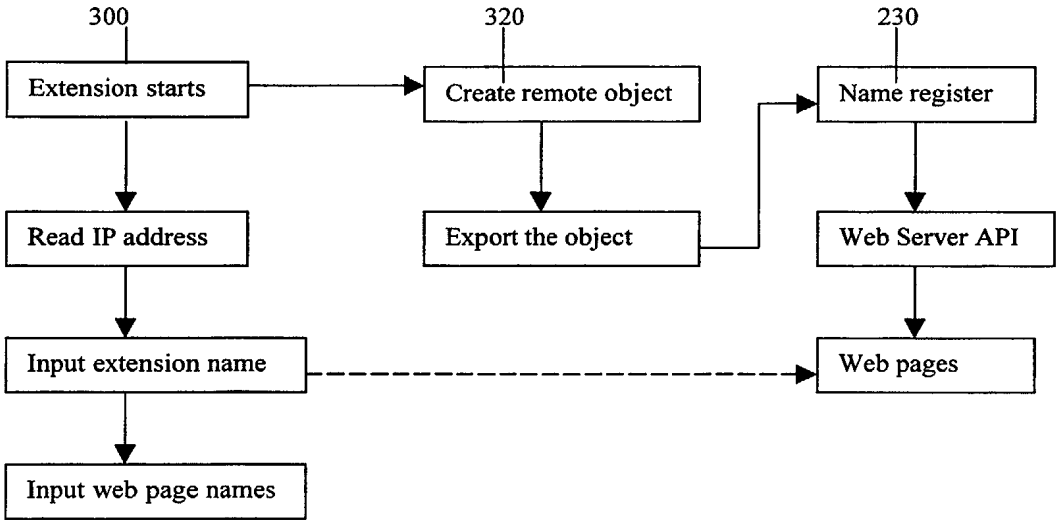

Fig. 2 illustrates the registration of a dynamic remote extension with web server

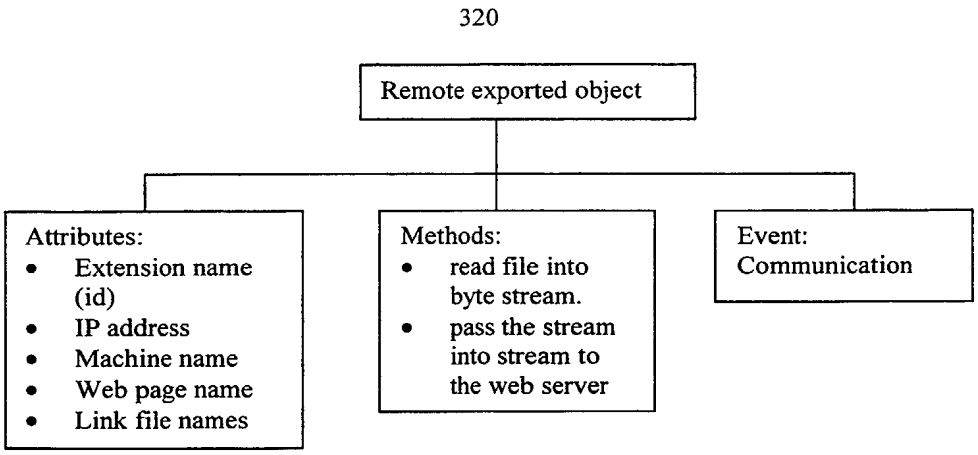

Fig. 3 illustrates the structure of exported object of a dynamic remote extension

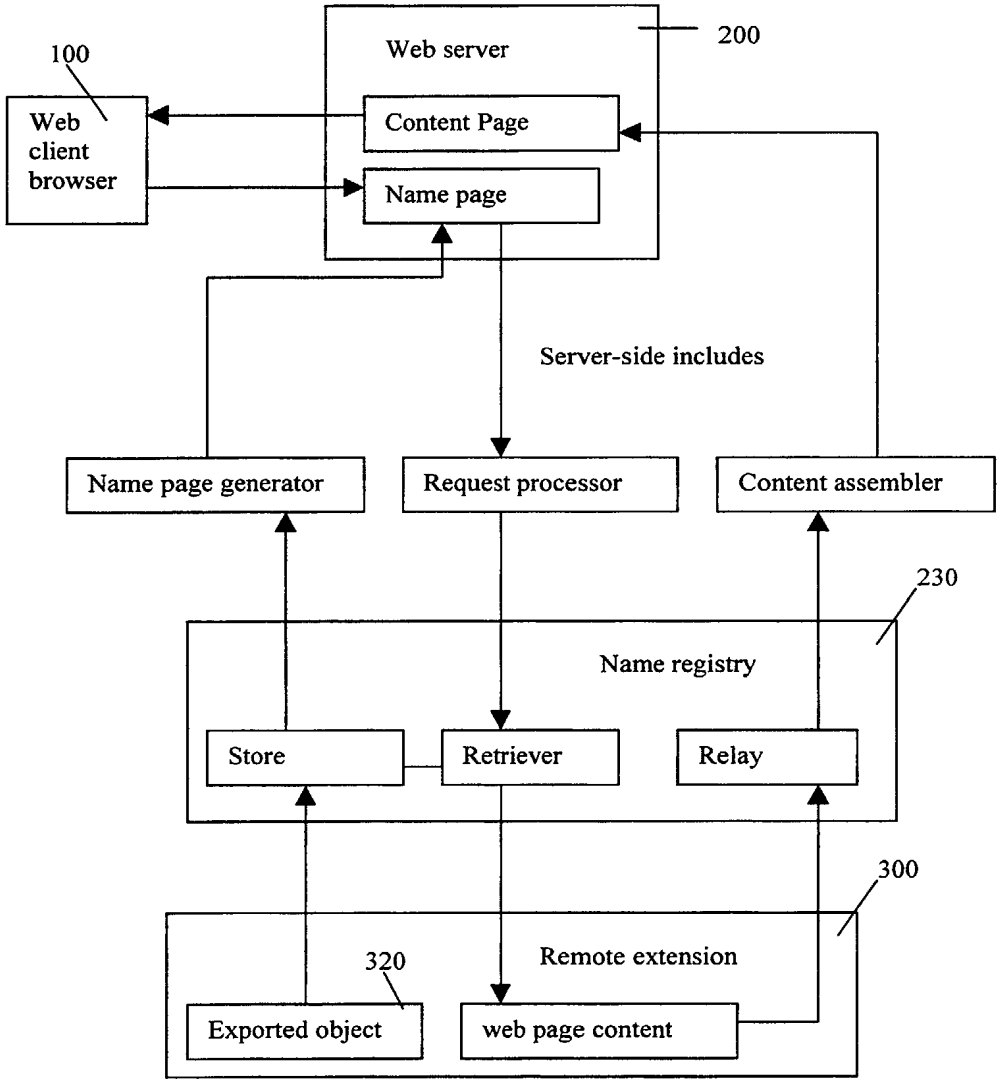

Fig. 4 illustrates the architecture and data flow of the web server network.

2369537

A Web Server with Dynamic Remote Extensions

The field of this invention relates to computer network software improvement for World Wide Web (web) server application, and more specifically, relates to dynamic web server construction without persistent IP address binding

Background of the Invention is to solve the problem of the dynamic binding of a web host name. This allows web users who haven't persistent IP address on the Internet to deliver their web service over the Internet.

An internet is a group of networks and individual computers communicating via a common protocol, namely the TCP/IP (Transmission Control Protocol/Internet Protocol) protocol suite. TCP/IP is a set of protocols and programs used to interconnect computer networks and to route traffic among different types of computers. A machine on the Internet can communicate with others following the TCP/IP protocols and the machine is referred to as a host or a node which is defined by their Internet (or IP) address. An IP address is a host identity on the Internet. The identity could be persistent one, i.e., for every time online a host uses a identical IP address, or could be dynamic one of which the online host may be changed from time to time.

A web server is a piece of software running on a host to provide web service to other hosts. To advertise a host in the Internet, the first thing is to register the host domain name with one of internet domain name service (DNS) servers that are organised as network hierarchy. Each DNS server has a file containing information such as names of hosts and their corresponding IP addresses, aliases for these hosts. A domain name consists of two parts: the actual host name, and the internet hierarchy to which it belongs. To find a host on the Internet, the domain name has to be resolved, i.e., to be translated into its corresponding IP address in a corresponding DSN server, because the internet defined communication socket only accepts a numeric IP address.

When a web client enters a domain name, in the form of uniform resource locator (URL), into a browser window, the browser queries its own local domain name server, which is probably located and maintained by a local Internet Service Provider (ISP) to translate the URL into IP address. If this IP address is on your own local network then the correct IP address will be returned and the browser will display the information requested. If, however, the web host is located out with the local network, then the chances are the IP address of the external resource will not be known by the local domain name server. In such instance, the local domain name server contacts the root domain server that managers information at higher level in the internet hierarchy. The information returned by the root domain server is the names of corresponding names of domain name servers, namely primary and secondary domain name servers. The local domain name server now contacts the corresponding servers to convert the URL into a IP address. The web browser then uses this IP address to contact the correct web host.

A domain name of a web host is bound to its IP address persistently and the both are stored in the containing file of a DNS server. This mechanism demands the web host has to have a persistent IP address. However, most of individual web surfers who connect to the Internet using the telephone dial-up line have not persistent IP address. The IP addresses are dynamically assigned by their ISPs during online connections. These IP

address are so called dynamic IP addresses. A web surfer has his IP address only if he is online. When the web surfer is offline, his IP address may be assigned to another online user. Therefore the IP address is subjected to be changed from time to time when the surfer connects to the Internet online. A dynamic IP address of a host cannot be resolved by a DNS server which needs to know the IP address of the host in advance. A conventional web server software also needs the IP address of the server host for configuration prior to its running. Web suffers cannot establish their web service on their own hosts without persistent IP addresses. If these surfers want to advertise their presence in the Internet, they have to go to chat room, white board, or to pay an ISP to house its content on the Web and make it available through the ISP's Web server, this is so called an virtual host. The virtual host is located on the server's host and these surfers have to pass their web contents to the server host via same protocols such as file transfer protocol (FTP). It would be difficult for these surfers to control their web contents and also encounter many limitations, such as space and content limitations.

An objective of this invention is to provide a solution of a web server with dynamic extensions, which allow web surfers without persistent IP address to deliver their web service through the Internet. It may be an alternative for the mechanism of domain name and IP address aliases.

In brief, the state of art of this invention is that a web server dynamically bind remote host web names with remote objects of these hosts to create web server extensions without IP address configurations. Instead of binding host web names with their low level IP addresses, the invention bind these names with high level remote objects that are exported by these remote hosts. These remote objects wrap host IP addresses and other system properties in run-time or online rather than prior configuration. Once a remote object is created, it wraps the host properties and then the remote object is exported to a web server. The web server will bind the host web name with the remote object and publish the host web name on its web page. This binding occurs only if a remote host is online and alive.

In accordance with this invention, a web server with remote extensions that only have dynamic IP address can be established. The web server acts as a switch board to connect these extensions hosts. Unlike virtual hosts, the resources of the web services are located in these extensions' hosts and these extensions can provide their own web service over the Internet. These extensions are dynamically online registered with the web server without IP configuration. If an extension is offline, it would be unregistered. An extension registration procedure requires the IP address of the extension in its run-time rather than prior time. Such an extension of the web server can use a dynamic IP address to deliver its web content over the Internet

The web server should have four tiers of a architecture: a HTTP server, a server-side includes (SSI), a name registry and remote extensions. The HTTP server houses the web pages that are dynamically create by SSI to publish the remote extension names and their related information. The main tasks of SSI are to create displayed web page and to process web clients' requests. The SSI is preferably the Java Servlets or CGI scripting. The name registry is used to store the extensions' link objects associated with their names and relay the web content object from extensions to SSI. The remote extensions house web contents that can be accessible over the Internet.

A name registry binds a host name with its exported object of which its IP address is one of attributes. This binding allow IP address to be bound in run-time. The name registry receives the both of the name and exported remote object. The registry stores the object in its memory in the terms of data structure, such as hash table or tree map, in association with the name. The name is also forwarded to SSI to generate a dynamic web page for its publish.

A web server extension is a dynamic remote host rather than a virtual host. The extension is a software which runs at remote host and makes the host become part of the web server. The IP address of the extension is encapsulated into a remote exported object in run-time and there is no configuration procedure to be carry out. When the extension runs on a remote machine, it encapsulates the following object items into the remote exported object: a.) the remote machine's properties, namely host machine name and IP address, b.) user input names and web page references, c.) methods that read the machine directive files, and d.) an event which keeps a communication channel alive in order to listen the oncoming requests. The extension exports the its object to the name registry and registers it with the registry.

A preferred embodiment of the invention will now be described with reference to the accompanying drawings in which

FIGURE 1. illustrates physical internet connection.
FIGURE. 2 illustrates the registration of a dynamic remote extension with web server
FIGURE. 3 illustrates the structure of exported object of a dynamic remote extension
FIGURE. 4 illustrates the architecture and data flow of the web server network

Detail Description
FIGURE. 1 illustrates an overview of the web server with dynamic remote extension network. The web server is a software package that runs on a physical host 200, which should be a high performance server machine. Extensions are also software packages that run on remote host 300, namely a PCs or workstations. Web clients 100 is a web browser to look for web pages.

The web server is a conventional web server with a persistent IP address and has registered its domain name with one of internet domain name server. The web server has two kinds of connections via the Internet: a HTTP connection which mainly connect to web clients for them to look up its web pages, and a self-specified protocol connection which is used to communicate with its extensions. The self-specified is non-HTTP communication protocol and could be primarily, though not exclusively, the Java remote method invocation (RMI), the Internet InterORB protocol (IIOP) of the common object request broker architecture (CORBA) , and the distribute common object model (DCOM). These protocols have to pass both object value and reference over the network. For instance, if use the CORBA's IIOP, the version of CORBA has to be 2.3 or above (the latest one is 2.3.1).

Fig 2 illustrates a procedure of a remote object registration. A remote extension 300 is a piece of software used to create a dynamic remote sub-server. The extension exports an object which encapsulates the host system information such as IP address with file reading behaviour and with the communication events to the name registry. The exported object 320 works as a stub to maintain a communication channel between name registry

and the extension host. The extension host provides file reading service to the name registry. It can read file located in the host on the behaviour of the name registry

The first main attribute of a exported object is the IP address of the its extension host. The IP address is read from system properties instead of from pre-configure files that may or may not exist in the host. The second attribute is the name of the extension host. The name can be specified by an owner of the extension or by the system predefined values. The name serves as a reference to the extension and it also is the identity of the extension (ID). It will be published on the web server's web pages and as a reference to retrieve the remote exported object in the name registry. The third attribute is the names of web pages that the owner of the extension want to publish on the Internet. These names are references to the owner's web contents. It should have at least one entry file name and may have other linked file names including image and audio file names. The entry file name could be a system default or be specified by its owner. The linked file names could be prompted by the owner or created by parsing the entry file name with system parser.

In addition to above attributes, the a remote object 320 also encapsulates read file behaviour and communication event. The structure of the remote object is shown in Fig. 3 and it conforms a standard object pattern consisting of attributes, methods and events. The attributes includes the extension name as the extension id, the IP address as the host ID, the name of the extension machine as second host ID though it may not be necessary, the name of web page as a web entry file name, and the linked file name to support the web page. The main methods in the object are read file into bytes and pass byte stream to remote host functions. The byte stream of an object is used to pass the object value to a remote host. A read file method is to read a file normally in textural or binary formats into a stream of bytes. Another method is to transfer this stream to a remote host. The exported object will work as a stub to keep a connection between the extension and the name registry. It generates a communication event and listen to the event taking place.

The remote object is exported to name registry for online registration. The name registry uses the extension name as a object identity to retrieve the object from its memory storage. The extension's name is then sent to the SSI which will generate a dynamic web page for publish the name, the dot line in Fig 2 indicates the publication. Once the extension started, the communication channel was established. The network is ready for processing the requests of web clients.

In the term of client/server , it can be describe as follows:

1.) First, the name registry would be a remote server and the extension be a client. The registry provide a registration service for the extension. The extension sends a request and exports a remote object for the registration.
2.) After registration, the role would be swapped , the extension would be a server which provides a read content, primarily web pages, service in a dedicate directory to the name registry. The registry is a client which forward the extension a request to read content from the SSI.

The main functions of a remote extension are to open a communication channel, to pass the host information to a name registry for registration, and to provide service to the registry.

Fig. 4 illustrates the architecture and data flow of a web server and its extensions. The web server 200 has multi-tiers architecture, the top tier is a HTTP web server which provides web page service for web clients to look up. It provides an user interface for web clients to interact. The second tier is a server-side includes SSI of the web server, such as common gateway interface CGI scripts or Java servlets. It acts as a web server application program interface (API) to run a computer program in responding to a coming request. This architecture is a conventional web application architecture.

In addition to the conventional web server, there is a tier called extension name registry 230. The registry has multiple roles in the terms of client/server. It provides name registration service to remote extensions 300 and processing request service to SSI. The name registry 230 at this stage is a server of both remote extensions 300 and SSI. After a remote extension 300 is registered, the name registry 230 becomes a client of the extension but still is a server of the SSI. The name registry 230 forwards the request from SSI to the extension and asks a service of the extension to read web contents residing the extension host.

The name registry 230 communicates with the SSI via a self-specified protocol and port number mentioned early. The name registry could reside the same machine as the web server resides but in different directories, or on different remote machines. The name register cannot directly be accessed by web clients 100. The communication protocol between the name register and remote extensions is also self-specified. Normally the name registry and the remote extensions are in different machine hosts.

This invention is a combination of the web server and web distribution computing and can be implemented using various technologies. The web server may be referred to open source Apache server, MS internet information service (IIS) server, Sun's JSP, or other commercial products. The web distribution computing technologies are primarily, not exclusively, the Sun's Java, Microsoft's DCOM, or OMG's CORBA. A practical instance of this invention using open source and Java technology is described in following sectors as an example to show how the invention is implemented.

The web serve is an Apache 1.3.12 server and the SSI is the Apache JServ1. 1. 2. The programming platforms are Sun's Java 2 version 1.3 and Java servlets development kit, JSDK2.0. The apache server is used to locate a web page which is dynamically created by a SSI object. The SSI has three servlet objects: a name page generator to publish the names of the server's extensions, a request processor to process the request from user interactive interface, and a content assembler to generate web pages based byte stream resulting from the request. The name registry 230 is a Java RMI-IIOP server and the request processor and remote extensions 300 are its RMI-IIOP clients. The communication port for RMI-IIOP is 3881.

When a web client 100 which is a web application namely a web browser sends the web server 200 a request, the web page generator of SSI retrieves the extensions names and their brief descriptions from the name registry 230. The generator generates a interactive web page with these name and descriptions. The web page works as a name list with name searching facility. The page has to be dynamically created because remote extensions 300 might be online or offline during the server is running. Once a remote extension is offline, its name shouldn't appear in the web page, whereas, if a new remote extension is online, its name should be added to the name list of the web page. The web

client 100 selects a name from the name list and sends the request processor of the SSI a connection request. The request processor invokes a method of the name registry 230 and pass the selected name in the term of a parameter to the name registry 230

The name registry 230 is a RMI-IIOP server which at least bound two remote object service via two remote interfaces: one is for SSI processor objects and another is for remote extension registration. Once a remote extension 200 is registered at the name registry, the name registry 230 becomes a RMI-IIOP client of the extension. There are many ways to do it: use RMI-IIOP call back mechanism, create a new RMI-IIOP server, or open a pair of communication sockets.

The name registry 230 retrieves an remote object 320 of a remote extension which is stored in the memory of the registry associated with the name parameter. The remote object contains the properties that indicate where the remote extension host resides on the Internet to resume the communication. The name registry 230 invokes a remote method of the extension object to read web page file on the remote host with file names as parameters that are stored in the remote object. The files on the remote host are read into low-level byte stream. These byte stream will be passed back to the name registry. The name registry forwards the byte stream to SSI. A content assembler object in SSL converts the byte streams to a web page format and then generate web pages, namely html or xml files. These web pages will sent back to the web clients in responds to its request. The web page of the extension 300 is, therefore, displayed on the web client 100's browser.

CLAIMS

What is claimed is:

1. A web server with dynamic remote extensions is a computer network with TCP/IP connection, comprising:

   A computer as a web server is used to dynamically bind its remote extensions that are computers in different locations. The server contains an extension name registry which registers the names (references) of its live remote extensions and binds these names with exported objects by these extensions. Unlike domain name service (DNS) which resolves a domain name to a IP address, the web server works as a gateway or an agent of a web clients. It receives the client's request and reads web contents from a required extension. These web contents are transferred into byte stream and sent to the web server via self-defined protocol such as, IIOP, RMI or CORBA. The web server reassembles the byte stream into web page format, namely HTML/XML files, and sends them back to the web client.

   Remote extensions can register itself with the web server in the extension's runtime by exporting its properties and behaviour as remote object to the web server. This registration only takes place in the runtime and it doesn't need a static IP address. This allows Dynamic Host Configuration Protocol (DHCP) users or others who haven't static assigned IP addresses to become parts of web server to publish their web contents on their own machine rather than on virtual hosts. They generally do not need to have addresses configured for DNS.

2. The web server with dynamic remote extensions of claim 1, wherein a remote dynamic extension software read runtime IP address and other host system properties information. These properties are encapsulated with web content references, content reading behaviours and communication listening events to form an object to be exported. The exported object is sent to the web server for registration. After registration, the remote extension becomes an independent web host of its web contents.

3. The web server with dynamic remote extensions of claim 1, wherein the web server binds lookup references with the exported objects of remote extensions. The web server stores remote objects of live extensions in its memory in the terms of data structure. It retrieves an exported object in responding to a client request and invokes the behaviours of the object to read web contents from the corresponding extension. On the other hand if a extension is offline, the corresponding remote exported object of the extension in the memory is eliminated and the extension is unregistered.

4. A web server with dynamic remote extensions substantially as herein described and illustrated in the accompanying drawings.

INVESTOR IN PEOPLE

Application No: GB 0028616.1 **Examiner:** Dr Albert Mthupha
Claims searched: 1-4 **Date of search:** 17 December 2001

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.S): H4P (PPG)

Int Cl (Ed.7): H04L (29/06)

Other: ONLINE : EPODOC, JAPIO, WPI.

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
A	GB 2338870 A	CHEN, see whole doicument.	
A	GB 2333670 A	TELEFONAKTIEBOLAGET, see whole document.	
A	WO 99/33211 A1	MEDIAONE, see whole document.	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.